98 P 1296

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 279 564 B1

R 7

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
10.07.1996 Bulletin 1996/28

(51) Int Cl.[6]: **G06F 13/42**

(21) Application number: **88300954.0**

(22) Date of filing: **04.02.1988**

(54) **Interface for use between a periphery and an asynchronous or synchronous system**

Schnittstelle für die Anwendung zwischen einer Peripherie und einem asynchronen oder synchronen System

Interface pour utilisation entre une périphérie et un système asynchrone ou synchrone

(84) Designated Contracting States:
AT BE CH DE ES FR GB GR IT LI LU NL SE

(30) Priority: **12.02.1987 US 13845**

(43) Date of publication of application:
24.08.1988 Bulletin 1988/34

(73) Proprietor: **ADVANCED MICRO DEVICES, INC.**
Sunnyvale, CA 94088 (US)

(72) Inventors:
- **Joshi, Sunil**
Campbell California 95008 (US)
- **Iyer, Venkatraman**
Berkeley California 94708 (US)

(74) Representative: **Wright, Hugh Ronald et al**
Brookes & Martin
52/54 High Holborn
London WC1V 6SE (GB)

(56) References cited:
EP-A- 0 135 879 **US-A- 4 379 327**

EP 0 279 564 B1

Description

Background of the Invention

The invention relates to the field of interfaces between processors and peripherals. More specifically, the invention relates to the field of processor to peripheral interfaces for processors which are either synchronous or asynchronous.

There are many applications today where microprocessors running various control programs are coupled to peripheral chips which are used by the microprocessor to do various functions necessary for the application. An example would be a UART for transmitting and receiving data over buses or a data path controller for a local area network application. Many other examples of peripherals exist, and new types of peripherals will be continuously developed. In these microprocessor applications, the peripherals have their own clocks which are asynchronous to the microprocessor clock. Typically, the microprocessor sends commands and data to the peripheral device and reads data from the peripheral device. Special interface circuitry is necessary to insure that these read and write transactions occur properly given that the two devices are running on different clocks. Because of this aspect and the fact that multiple peripherals may be connected to the same data bus, there exists a need for a predetermined set of handshaking signals to indicate to a peripheral when it is being addressed, what kind of a transaction is supposed to occur, when the peripheral is ready, and when the data is on the bus.

Other applications use bit sliced microprocessors or microprogrammed architecture coupled to peripherals all of which run on the same system clock. In these architectures, the controller sends instructions to the peripheral on an instruction bus and sends an instruction enable signal to the peripheral to enable decoding of the instruction. The peripheral is also coupled to the system data bus which is also coupled to the ALU of the microprogrammed system. In these systems, the peripherals must have a special interface to decode the instructions, tri state the data bus and enable the proper register or other logic in the peripheral that the controller wishes to read from or write to.

Heretobefore, the interfaces needed by the peripherals in each of these two types of systems needed to be different because of the different handshaking signals involved to synchronize the transfer of data because of the different situations with respect to the clock and the different transaction methods in the two cases. The two types of systems often use the same types of peripherals. Manufacturers of peripherals therefore must manufacture two different types of peripherals which do the same function but which have different interfaces. This causes extra expense.

Thus there has arisen a need for a universal interface which may be used in both asynchronous and synchronous applications.

From EP-A-135879 a system is known wherein a memory select signal and an address signal are combined with a local clock signal to provide an instruction enable signal to a memory controller if the "asynchronous mode" signal is activated; if the "synchronous mode" signal is activated the access request signals are gated through.

The present invention provides an interface to allow a peripheral device to communicate with either synchronous or asynchronous systems comprising first means for combining and synchronizing a data strobe and a chip select signal from an asynchronous system and converting the combined signal into an instructions enable signal when a mode signal is in a first state: means for gating an instruction enable signal from a synchronous system through as the peripherals internal instruction enable signal when said mode is in a second state: characterised in that second means is provided for combining one signal on an instruction bus from either a synchronous or an asynchronous system with said combined data strobe and chip select signals to provide a read/write signal for said peripheral.

Brief Description of the Drawings

Figure 1 is a logic diagram of a typical microprogrammed synchronous processor-peripheral system.

Figure 2 is a logic diagram of the logic inside a typical peripheral used in the invention.

Figure 3 is a timing diagram of a typical data exchange operation in the system of Figure 1.

Figure 4 is a logic diagram of a typical asynchronous processor-peripheral interface.

Figure 5 is a timing diagram of a typical data exchange in tne system of Figure 4.

Figure 6 is logic diagram of the universal interface according to the teachings of the invention.

Figure 7 is a timing diagram of a data exchange between a processor and a peripheral using the interface using the interface of Figure 6.

Figure 8 is a timing diagram for read and write cycle timing for transactions with an asynchronous processor.

Figure 9 is a timing diagram for read and write transactions with a synchronized processor.

Figure 10 is a connection diagram for connection of the interface to an asynchronous processor.

Figure 11 is a connection diagram for connection of the interface to a synchronous processor.

Detailed Description of the Preferred Embodiments of the Invention

Figure 1 shows a block diagram of a typical synchronous architecture in the form of a microprogrammed system. A sequencer 20 generates addresses on a bus 22 in response to clock signals on the system clock bus 24. These addresses are sent to a microprogrammed PROM or ROM 26 where they cause access of a particular microcode word stored at that address in the PROM 26. Typically, the microcode word is one word in a sequence of microcode words which together comprise a control program which causes the system to perform the function desired by the user.

The control word emerges from the PROM 26 on a data bus 28 and is clocked for storage into a pipeline register 30. The control word is comprised of a plurality of fields, each field being a control signal or instruction field for one of the peripherals in the system. In the system of Figure 1, only two peripherals are shown, an ALU 32 and a generic peripheral 34. Each peripheral receives its instructions from the register 30 via an instruction bus dedicated to that peripheral and carrying a plurality of control bits derived from some field in the control word stored in the register 30. In the case of the ALU 32, the instruction bus 36 carries an N bit instruction word having the same binary bode as the N bits of the ALU instruction field in the control word stored in the register 30. Likewise, the peripheral 34 receives a 5 bits (typically 5 bits but the instruction word could have other numbers of bits as well) on an instruction bus 38.

The instruction word tells the peripheral whether the transaction desired by the controller is a read or write and which register in the peripheral is to be written to or which register is to put its data onto the system data bus. Typically, the microcode word will control a number of peripherals simultaneously. When a transfer is to occur, it will be between two different peripherals and these two peripherals will be addressed. One peripheral will instructed to place the contents of one its registers on the system data bus and another peripheral will be instructed to take the data on the data bus and store it in one of its registers.

Another bit of the register 30 will be used for each peripheral to implement the instruction enable signal IEN* (an * following a signal name indicates an active low signal indicated by an overscore on the drawings). The instruction enable signal is a signal which tells the peripheral that the instruction on the instruction bus for that peripheral is to be decoded, and carried out, i.e., that the peripheral is being addressed. In the case of the peripheral 34, the instruction enable signal is the signal IEN1* on the line 40. In the case of the ALU peripheral 32, the instruction enable signal is the signal IEN2* on the line 42.

Figure 2 shows the internal details of the typical peripheral and how the instruction enable selects the internal decoder and causes the decoder outputs to be activated to select the desired internal register or other logic of the peripheral which is to be read or written. Figure 2 is shown as a typical internal detail for a peripheral such as peripheral 34. It is not intended to represent the exact internal detail of any particular peripheral or all peripherals in general. Figure 2 is intended only to illustrate the use of the interface signals in a typical synchronous microprogrammed system for purposes of framing the context of the invention. Typical operation of a peripheral is as follows.

The instruction indicating which register inside the peripheral that is to be addressed arrives on the instruction bus 38 and is decoded by a decoder 50. This decoding happens only when the I EN*1 enable signal is active low. The decoder decodes the bits on the bus 38 and activates one of the control signals on the bus 52.

The bus 52 includes two different types of signals. There are a plurality of select signals on the bus 54 and individual load signals for each of a plurality of registers of which only two are shown. The shared system data bus 56 is coupled directly to all the data inputs of the registers and is coupled through a tri state gate and multiplexing system to all the data outputs of the registers. The individual load signals such as the signals on the lines 58 and 60 are coupled to the load control inputs of their respective registers. When any particular load signal is active, the associated register is loaded with whatever data is on the system data bus 56. All the other data inputs are high impedance and do not load the system bus 56.

The select signals on the SELECT signal bus 54 are coupled to the select inputs of a multiplexer 58. The multiplexer 58 has a data input coupled to the data output of each register. Each select input signal, when activated, causes the associated data input to be coupled to the single data output 62. This single data output 62 is coupled through a tri state gate 64 to the system data bus 56. The tri state gate 64 maintains its output coupled to the system data bus 56 in a high impedance state at all times when the READ signal on line 66 from the decoder 50 is not active. READ is activated by the decoder 50 when any of the signals on the select bus 54 are active.

Figure 3 is a timing diagram of a typical read or write transaction to the peripheral 34 in Figure 1. For simplicity, only the rising edges of the system clock are shown. At clock transition time t0, the instruction bus bits assume their states for either a read or write instruction. At the same time t0, the IEN*1 signal goes low which causes the decoder inside the peripheral to decode the instruction. Also, at time t0, or near that time, the data bus is driven with the data to be written to the peripheral. When the enable signal goes low, the proper register load input addressed by the instruction bus word is driven low and the data on the data bus is loaded into the register. In the case of a read instruction, the enable signal causes the decoder to decode the instruction and the proper select signal on the SELECT bus 54 is generated to cause the output of the proper register in the peripheral to be selected and applied to the data bus. Also, the enable signal cause the READ signal on line 66 to be activated so as to cause the tri state gate 64 to gate the

selected output onto the system data bus 56. The data is then latched into a register in some peripheral or other unit connected to the system data bus which has been given an instruction to receive the data. This manner of peripheral information exchange is known in the prior art for microprogrammed systems.

Referring to Figure 4 there is shown a typical microprocessor based system using a peripheral having a port for receiving address signals defining which internal register is to used in a read or write transaction. In the system of Figure 4, the microprocessor 70 has a multiplexed address/data bus 72. In other embodiments, there may be separate address and data buses. Figure 5 is a timing diagram of a typical data transaction between the microprocessor and the peripheral 74 having multiple internal registers. Joint reference to Figures 4 and 5 will be used to explain the typical operation of the system of Figure 4.

Unlike the microprogrammed system of Figure 1, the microprocessor and the peripheral in the system of Figure 4 run on separate clocks. This means that a handshake must be used in the system of Figure 4 to synchronize the proper transfer of data in contrast to the synchronous, single clock cycle operation of the system of Figure 1. The details of this handshake or protocol are shown in Figure 5. Assume that the microprocessor is to send a data word to one of the internal registers of the peripheral. The first step is to address the peripheral and tell it which of the internal registers is to receive the data. In the system of Figure 4 one embodiment of how this is done is shown although there at least two other known ways of doing this in the prior art. Because the address and data bus are multiplexed, the bus 72 is first used to send the address of the peripheral and the particular register in that peripheral beginning at time t0. At time t1, an address latch enable signal ALE is activated. This signal is coupled to the load input of an address latch 76, and causes it to load the address on the bus 72. This address is coupled on a bus 80 to the inputs of a decoder 78 where it is decoded. The decoder activates a chip select signal CS* on a line 82 coupled to the chip select input of the peripheral 74. In the embodiment shown in Figure 4, if the address on the bus 72 was the address of any of the multiple registers in the peripheral 74, the decoder would decode the most significant bits of the address and this fact would be detected. If this turned out to be the case, then the decoder 78 would activate the CS* signal, and the peripheral would be activated.

The peripheral must be informed which of its internal registers to load or which register is to drive its contents onto the data bus 72 at the appropriate time. There are at least three ways of doing this. One way is to do the transaction in two cycles. In the first cycle, the identification of the desired register is sent to the peripheral and stored. In the second cycle, the register identified in the first cycle is either loaded with the data word from the microprocessor or the identified register drives its contents onto the data bus 72. The second way is to cycle through the registers in the peripheral sequentially so that on the first transaction, the first internal register is used, and on the second transaction, the second internal register is used. This scheme requires use of a software or hardware counter to keep track of which register is currently active. In the embodiment of Figure 4, the n least significant bits of the bus 80 are coupled to a port of the peripheral 74, and the bits on this bus 84 are decoded by the peripheral to determine which of the internal registers is to be accessed. The decoder 78 activates CS* if the address on the bus 80 is any one of the internal registers of the peripheral. The address latch 76 is needed only because the bus 72 is time division multiplexed with address an data information. If a separate address bus exists, the address latch 76 is not necessary.

The transaction is completed as follows. When ALE is activated to latch the address at time t1, the microprocessor also activates an R/W* signal to the proper state to indicate whether the desired transaction is read or a write. Figure 5 shows a write transaction, but a read transaction is similar. The address is then decoded, and chip select CS* goes low at time t2. The microprocessor then drives the data bus 72 with the data to be written as shown at time t3. Shortly thereafter, DS* is driven low by the microprocessor as shown at 90. The activation of data strobe causes the peripheral 74 to latch the data into the internal register indicated by the microprocessor either by the bit pattern on the bus 84 or by the register indicated by either of the other methods described above.

The latching of the data causes the READY* signal to be activated by the peripheral. This indicates to the microprocessor that the data has been received, so the microprocessor deactivates the data strobe signal DS* as shown at 92. This causes the READY* signal to be deactivated as shown at 94 and causes the microprocessor to remove the data from the data bus 72.

Because of the mixed nature of systems and peripherals that exist in the marketplace, a need has arisen for a processor peripheral interface which can serve both the synchronous and asynchronous markets for microprogrammed systems and microprocessor systems, respectively. Obviously, the pins and signals are different but there is some overlap in function which allows a naming convention to be adopted and a sharing of signals by a universal interface with the addition of a single signal which indicates to the interface whether it is to operate in the synchronous or asynchronous mode. The correspondence between the signals which are used in the interface according to the teachings of the invention is as follows.

microprogrammed synchronous	microprocessor asynchronous
INSTO-N	=AO - AN (N LSBs on bus 84)

(continued)

microprogrammed synchronous	microprocessor asynchronous
IEN*	=CS*
MSB of INST	=R/W* for N = 4, i.e., the R/W signal is INST bit 4
DS* not needed	
READY not needed	
BMODE	

The definitions of these signals for the interface of the invention is INST0-N is the instruction bus. These bits are decoded with the peripheral to select the internal register or to perform certain functions. If the number of read and write instructions are equal, INST4 can become the equivalent of the R/W* pin and INST0-3 can select the internal register. IEN* is the enable pin which lets the instruction be executed. It is equivalent to a chip select signal. The DS* signal from a microprocessor using the interface can be directly connected to the interface. In the case of a microprogrammed system, DS* can be connected to the IEN* line externally or permanently wired to a logic 0 level. READY* from the peripheral to the microprocessor that sends the data to signal that the transfer is complete becomes active when the data is properly strobed. It is not needed in synchronous systems. The BMODE signal, when high, indicates that the processor runs synchronously to the peripheral. When this signal is low, the processor runs asynchronously. A command register bit may be used in place of the pin.

Figure 6 is a logic diagram of the logic of the peripheral interface of the invention which will allow a peripheral using the interface to do data exchange with either a synchronous or asynchronous processor. Figure 7 is a timing diagram describing the operation of the interface of Figure 6. Referring jointly to Figures 6 and 7, the operation of the interface of Figure 6 will be given.

The interface serves to perform a process of clock synchronization. The process starts when both the signals CS* and DS* on lines 100 and 102 become active low indicating that this particular peripheral is being addressed and that the data on the data bus is valid. The inverters 104 and 106 invert the signals, and the AND gate 108 signals the convergence by driving the D input of a first flip-flop 110 high. This event sets the Q output at the next upward transition of the PCLK signal on the line 114. The Q output of the flip-flop 110 is coupled to the D input of a second flip-flop 112 which reclocks the event to the next upward transition of the PCLK signal to avoid metastable states.

The third flip-flop 116 combined with the inverter 118 and the AND gate 120 serve as a synchronous one shot that generates a pulse at the rising edge on the line 122. The pulse width is equal to the delay through the flip-flop plus the delay through the inverter 118 and the gate 120 plus any delay between the rising edge on the line 122 and the next rising edge of PCLK. The pulse output from the gate 120 on the line 122 serves as the IEN* signal for the peripheral enabling decoding of the instruction encoded on the INST0-4 bus 124. Of course, the BMODE signal on line 126 must be high indicating peripheral is designed to work in a synchronous system. When the BMODE signal is high, the multiplexer 126 must select the signal on the line 122 for coupling to the line 128 as the signal INSTEN. This signal is coupled to one input of each of the AND gates gating the outputs of the decoder 130 which has decoded the instruction on the bus 124. Thus when INSTEN is high, the decoded instruction is sent out on the bus 132 to the circuitry in the peripheral which is to be controlled by the instruction. Thus a peripheral which is designed to work with a synchronous microprogrammed system which generates a signal IEN* may be made to work with an asynchronous microprocessor which does not generate IEN* but does generate a CS* and a DS* signal.

When the peripheral is designed to be coupled to an asynchronous microprocessor and is expecting a DS* and a CS* signal, the BMODE signal is low which causes the multiplexer 126 to select the signal on the line 134. Line 134 goes high when both CS* and DS* are active. When line 134 goes high and BMODE causes multiplexer 126 to select the line 134, then INSTEN goes high. This causes the instruction on the INST0-3 bits on bus 136 to be decoded and passed through the AND gates at the output of the decoder 130 to the logic of the peripheral via the bus 132. This parallels normal asynchronous operation with the INST0-3 bits substituting as the address bits on the bus 84 in Figure 4. Of course the DS* and CS* signals serve their normal functions as gated and guided by the circuitry of Figure 6.

Because a R/W* signal is needed for asynchronous operation, the INST4 is used as this signal by breaking it out separately on the line 138. This signal is gated with the signal on line 134 in the AND gate 140 to generate the DBEN signal on the line 142. When the peripheral is selected and DS* is active this indicates that the data bus is being driven with data from the microprocessor or that the microprocessor is ready to receive data. The data bus will then be handled by the peripheral in accordance with the state of the the signal on the line 138. If the transaction is to be a read, the signal on the line 138 will be high as will the signal on the line 142. This will cause the data bus tri state driver 144 to drive the data bus lines DB15-0 with the data on the DBOUT15-0 lines of bus 146. If the transaction is to be a write, the bus receiver 148 will drive the data bus inside the peripheral, DBIN15-0, with the data on the data bus 150. The tri state bus driver 144 will be in a high impedance state thereby isolating the data outputs of the data registers in the

peripheral from the data bus 150.

The READY* signal needed for asynchronous operation generated by the peripheral for transmission to the microprocessor indicating that the peripheral has received the data is generated by the peripheral interface from the output of the inverter 116. As seen from the timing diagram of Figure 7, it is seen that when the DS* signal goes low, the READY* signal goes low after some propagation delays. When the READY* signal is received by the microprocessor, the DS* signal is deactivated by the microprocessor as shown in Figure 7 at 152. This causes the READY* signal to again go high at 154 after the change in DS* to a logic 1 causes the signal on line 134 to go low thereby clearing all the flip-flops in Figure 6.

The gate 156 serves to prevent an anomaly in decoding the chip select signal from causing the wrong peripheral to clock in data not intended for it. Sometimes the decoder outside the peripheral which decodes the chip select signal CS* (not shown) from the address bus is slow enough in decoding the CS* signal that the ultimate result does not occur until after the DS* signal goes low. This condition is shown in dotted lines in Figure 7 as the interval 160. During this interval, the chip select signal CS* may be going up and down as the decoder works its way through to the ultimate outcome. It is possible that the DS* signal will go low at 162 at a time when the CS* signal is low. It is also possible that the ultimate outcome of the decoding of the address will be as shown at 164 and the address is not that of the peripheral. If this occurs, then data will be clocked into the peripheral that is not intended for it. To prevent this from occurring, the gate 156 is used. The purpose of this gate is to block clocking of the signal on the line 134 through the two flip-flops 110 and 112 into the flip-flop 116 unless the condition of the signal on the line 134 still persists two clock cycles later. That is, if CS* and DS* both go low, the signal on the line 134 will go high. This logic 1 will be clocked into flip-flop 110 on the next clock pulse, and will be clocked into the flip-flop 112 on the second successive clock cycle. However, the AND gate 156 will prevent the clocking of the signal at the output of the flip-flop 112 into the flip-flop 116 unless the signal on the line 172 is still logic 1 on the third successive clock pulse after the transition 162 in Figure 7.

Figure 7 illustrates the relative time durations of the INSTEN signal and the DS* signal. The figure shows that INSTEN is much shorter than DS*. This indicates that the microprocessor is expecting the data being driven on the data bus by the peripheral to be valid much longer than the INSTEN signal will enable the register driving the data bus. To insure that the data being read by the microprocessor is still there when the microprocessor is ready to read it, a pipeline register or storage register 166 is provided in the read path. This register is always enabled and latches the data driven on the bus 146 when INSTEN enables the register being addressed. The data is then preserved and is still present on the data bus 146 for driving onto the main data bus 150 even after INSTEN goes inactive.

Referring to Figure 8 there is shown a timing diagram of a write cycle for an asynchronous processor when BMODE is low. Note that even though the system clock driving the microprocessor is about 1/2 the speed of the peripheral clock signal PCLK, a read cycle can occur because of the connection of the line 142 and the action of the pipeline register. That is, the pipeline register latches the data which is driven onto bus 146 by the addressed register (not shown). This is true even though the register is enabled by INSTEN for only the single cycle of PCLK that INSTEN is true. Meanwhile, the DS* signal keeps the bus 146 connected to the bus 150 for the entire time the microprocessor is expecting the data to be valid, i.e., during the entire time the DS* signal is active. The latter fact is true because both CS* and DS* are low during the entire read data period, so line 134/172 is high for the entire read data period. If INST4 is high during the entire read data period, as it is normally, then gate 140 will drive the bus driver 144 to a non tri state condition through line 142 during the entire read data period. Figures 8 and 9 clearly show that the interface shown in Figure 6 can be used to for read or write cycles with either synchronous or asynchronous systems. Note the sequencing of the READY signal with DS* in Figure 8 and the correspondence of the timing with the timing shown in Figure 5.

Figures 10 and 11 show how a peripheral with the interface shown in Figure 6 could be attached to an asynchronous microprocessor system and a synchronous microprogrammed system. The various connections of the BMODE, clock and DS* signals for the two different configurations are shown in these figures for the necessary voltage conditions necessary for the interface to work in the two different configurations.

Claims

1. An interface to allow a peripheral device to communicate with either synchronous or asynchronous systems comprising:

   first means (108,110,112,156,116,118,120,122) for combining and synchronizing a data strobe (DS) and a chip select (CS) signal from an asynchronous system and corverting the combined signal into an instructions enable signal (INSTEN) when a mode signal (BMODE) is in a first state;
   means (108.122) for gating an instruction enable signal (IEN) from a synchronous system through as the peripherals internal instruction enable signal (INSTEN) when said mode (BMODE) is in a second state:
   characterised in that

second means (140) is provided for combining one signal (INST4) on an instruction bus (124) from either a synchronous or an asynchronous system with said combined data strobe (OS) and chip select (CS) signals to provide a read/write siqnal (R/W) for said peripheral.

2. The apparatus as claimed in claim 1 wherein said first combining means comprises

an AND gate (108) having a first input coupled through an inverter (104) to receive said chip select signal (CS) anc having a second input coupled through an inverter (106) to receive said data strobe signal (DS) and having an output:
synchronizing means (110,112) coupled to the output of said AND gate (108) for synchronizing the combined signal with a clock signal (PCLK) used by said peripheral and having an output:
means (116,118,120) coupled to the output of said synchronizing means for converting the synchronized version of the combined chip select and data strobe signals into a pulse;
means (126) coupled to receive said pulse and gate it through for use as the peripherals instruction enable signal (INSTEN) when said mode signal (BMODE) is in said first state.

3. The apparatus as claimed in claim 2 wherein said means (126) for gating comprises multiplexer (126) having a first input couoled to receive said pulse and having a second input coupled to the output of said AND gate (108) and having a select input coupled to receive said mode signal and providing said instruction enable signal (INSTEN) for said peripheral an its output.

4. The apparatus as claimed in claim 2 or 3 wherein said svnchronizing means (110,112) comprises a first D type flip-flop (110) having a D input coupled to said output of said AND gate (108) and having a clock input coupled to receive the peripheral clock signal (PCLK), and a second D type flip-flop (112) having a D input coupled to the Q output of said first flip-flop (110) and having a clock input coupled to receive said peripheral clock signal, the Q output of said second flip-flop being said synchronized, combined data strobe and chip select signal, and where both said first and second flip-flops have their clear inputs coupled to the output of said AND gate.

5. The apparatus as claimed in claim 2,3 or 4 wherein said mean (110,118,120) for converting the combined, synchronized signal comprises a third D type flip-flop (116) having its D input coupled to receive the combined, synchronized data strobe and chip select signals and having Its clock input coupled to receive the peripheral clock signal and having its clear input coupled to the output of said AND gate (108) and a second AND gate (120) having a first input coupled to the Q output of said flip-flop (116) and having its second input coupled to the D input of said flip-flop (116).

6. The apparatus as claimed in claim 5 wherein said means (126) for gating is a multiplexer (126) having its first input coupled to the output of said third D-type flip-flop (116) and having a second input coupled to the output of said first mentioned AND gate having its inputs coupled to receive said data strobe (DS) and said chip select (CS) signals, and having its select input coupled to receive said mode signal (BMODE).

7. The apparatus as claimed in claim 5 wherein there is provided means for using the gated signal as the peripherals read/write control signal to control the peripherals access to the data bus used by the peripheral to communicate with said system.

Patentansprüche

1. Interface zum Ermöglichen der Kommunikation einer Peripherievorrichtung mit entweder synchronen oder asynchronen Systemen, mit:

einer ersten Einrichtung (108,110,112,156,116,118,120,122) zum Kombinieren und Synchronisieren eines Daten-Strobe-Signals (DS) und eines Chipauswahl-Signals (CS) von einem asynchronen System und zum Umwandeln des kombinierten Signals in ein Befehlsfreigabesignal (INSTEN), wenn ein Modussignal (BMODE) in einem ersten Zustand ist;

einer Einrichtung (108,122) zum Durchlassen eines von einem synchronen System kommenden Befehlsfreigabesignals (IEN) als das interne Peripheriebefehlsfreigabesignal (INSTEN), wenn das Modussignal (BMODE) in einem zweiten Zustand ist:

dadurch gekennzeichnet, daß

eine zweite Einrichtung (140) vorgesehen ist, um ein Signal (INST4) auf einem Befehlsbus (124) von entweder einem synchronen oder einem asynchronen System mit den kombinierten Daten-Strobe- (DS) - und Chipauswahl- (CS) -Signalen zu kombinieren, um für die Peripherie ein Schreib/Lese-Signal zu liefern.

2. Vorrichtung nach Anspruch 1, bei der die erste Kombinationseinrichtung aufweist

   ein UND-Gatter (108) mit einem über einen Inverter (104) gekoppelten ersten Eingang zum Empfang des Chipauswahlsignals (CS) und einem über einen Inverter (106) gekoppelten zweiten Eingang zum Empfang des Daten-Strobe-Signals (DS) und mit einem Ausgang;

   eine Synchronisationseinrichtung (110,112), die mit dem Ausgang des UND-Gatters (108) gekoppelt ist, um das kombinierte Signal mit einem von der Peripherie verwendeten Taktsignal (PCLK) zu synchronisieren, und einen Ausgang aufweist;

   eine Einrichtung (116,118,120), die mit dem Ausgang der Synchronisationseinrichtung gekoppelt ist, um die synchronisierte Version der kombinierten Chipauswahl- und Daten-Strobe-Signale in einen Impuls umzuwandeln;

   eine Einrichtung (126), die gekoppelt ist, um den Impuls zu empfangen und zwecks Verwendung als Peripheriebefehlsfreigabesignal (INSTEN) durchzulassen, wenn das Modussignal (BMODE) in dem ersten Zustand ist.

3. Vorrichtung nach Anspruch 2, bei der die Einrichtung (126) zum Durchlassen einen Multiplexer (126) aufweist, der einen zum Empfang des Impulses gekoppelten ersten Eingang und einen mit dem Ausgang des UND-Gatters (108) gekoppelten zweiten Eingang und einen zum Empfang des Modussignals gekoppelten Wähleingang aufweist und an seinem Ausgang das Befehlsfreigabesignal (INSTEN) für die Peripherie liefert.

4. Vorrichtung nach Anspruch 2 oder 3, bei der die Synchronisationseinrichtung (110,112) ein erstes D-Flipflop (110), das einem mit dem Ausgang des UND-Gatters (108) gekoppelten D-Eingang und einen zum Empfang des Peripherietaktsignals (PCLK) gekoppelten Takteingang aufweist, und ein zweites D-Flipflop (112) aufweist, das einen mit dem Q-Ausgang des ersten Flipflops (110) gekoppelten D-Eingang und einen zum Empfang des Peripherietaktsignals gekoppelten Takteingang aufweist, wobei der Q-Ausgang des zweiten Flipflops das synchronisierte kombinierte Daten-Strobe- und Chipauswahlsignal ist, und wobei die Löscheingänge sowohl des ersten als auch des zweiten Flipflops mit dem Ausgang des UND-Gatters gekoppelt sind.

5. Vorrichtung nach Anspruch 2, 3 oder 4, bei der die Einrichtung (116,118,120) zum Umwandeln des kombinierten synchronisierten Signals ein drittes D-Flipflop (116) aufweist, dessen D-Eingang zum Empfang der kombinierten synchronisierten Daten-Strobe- und Chipauswahlsignale gekoppelt ist und dessen Takteingang zum Empfang des Peripherietaktsignals gekoppelt ist und dessen Löscheingang mit dem Ausgang des UND-Gatters (108) gekoppelt ist, und bei der ein erster Eingang eines zweiten UND-Gatters (120) mit dem Q-Ausgang des Flipflops (116) gekoppelt ist und sein zweiter Eingang mit dem D-Eingang des Flipflops (116) gekoppelt ist.

6. Vorrichtung nach Anspruch 5, bei der die Einrichtung (126) zum Durchlassen ein Multiplexer (126) ist, dessen erster Eingang mit dem Ausgang des dritten D-Flipflops (116) gekoppelt ist und bei dem ein zweiter Eingang mit dem Ausgang des erstgenannten UND-Gatters gekoppelt ist, wobei dessen Eingänge zum Empfang der Daten-Strobe- (DS) - und der Chipauswahl-(CS)-Signale gekoppelt sind, und dessen Wähleingang zum Empfang des Modussignals (BMODE) gekoppelt ist.

7. Vorrichtung nach Anspruch 5, bei der eine Einrichtung zur Verwendung des durchgelassenen Signals als Peripherie-Schreib/Lese-Steuersignal zum Steuern des Zugriffs der Peripherie auf den von der Peripherie zum Kommunizieren mit dem System verwendeten Datenbus vorgesehen ist.

Revendications

1. Interface pour permettre à un dispositif périphérique de communiquer soit avec un système synchrone soit avec

un système asynchrone comprenant :

- des premiers moyens (108, 110, 112, 156, 116, 118, 120, 122) pour combiner et synchroniser des données d'échantillonnage (DS) et un signal de sélection de circuit (CS) d'un système asynchrone et convertir le signal combiné en signal de validation d'instructions (INSTEN) quand un signal de mode (BMODE) est dans un premier état;
- des moyens (108, 122) pour sélectionner par une porte un signal de validation d'instruction (IEN) d'un système synchrone comme signal de validation d'instruction interne des dispositifs périphériques (INSTEN) quand le mode (BMODE) est dans un second état :

caractérisé en ce qu'un second moyen (140) est disposé pour combiner un signal (INST4) sur un bus d'instruction (124) venant soit d'un système synchrone, soit d'un système asynchrone avec lesdits signaux de données d'échantillonnage (DS) et de sélection de circuit (CS) afin de fournir un signal lire/écrire (R/W) au dispositif périphérique.

2. Appareil selon la revendication 1, dans lequel le premier moyen de combinaison comprend :

une porte ET (108) ayant une première entrée couplée par un onduleur (104) de manière à recevoir le signal de sélection de circuit (CS), une seconde entrée couplée par un onduleur (106) de manière à recevoir le signal de données d'échantillonnage (DS) et une sortie:
des moyens de synchronisation (110, 112) couplés à la sortie de la porte ET (108) pour synchroniser le signal combiné avec un signal d'horloge (PCLK) utilisé par le dispositif périphérique et ayant une sortie :
des moyens (116, 118, 120) couplés à la sortie du moyen de synchronisation pour convertir la version synchronisée des signaux combinés de sélection de circuit et de données d'échantillonnage en impulsion;
un moyen (126) couplé pour recevoir l'impulsion et la sélectionner par une porte afin de l'utiliser comme signal de validation d'instruction des dispositifs périphériques (INSTEN) quand le signal de mode (BMODE) est dans son premier état.

3. Appareil selon la revendication 2, dans lequel le moyen (126) de sélection par une porte comprend un multiplexeur (126) possédant une première entrée couplée de façon à recevoir l'impulsion, une seconde entrée couplée à la sortie de la porte ET (108) et une entrée de sélection couplée de manière à recevoir le signal de mode et fournissant à sa sortie le signal de validation d'instruction (INSTEN) au dispositif périphérique.

4. Appareil selon les revendications 2 ou 3, dans lequel le moyen de synchronisation (110, 112) comprend une première bascule de type D (110) ayant une entrée de type D couplée à la sortie de la porte ET (108) et une entrée d'horloge couplée de manière à recevoir le signal d'horloge du dispositif périphérique (PCLK), et une seconde bascule de type D (112) ayant une entrée de type D couplée à la sortie de type Q de la première bascule (110) et une entrée d'horloge couplée de manière à recevoir ledit signal d'horloge du dispositif périphérique, la sortie de type Q de la seconde bascule étant un signal synchronisé combiné de données d'échantillonnage et de sélection de circuit, et dans lequel la première et la seconde bascule ont leurs entrées libres couplées à la sortie de la porte ET.

5. Appareil selon les revendications 2, 3 ou 4, dans lequel ledit moyen (116, 118, 120) pour convertir le signal synchronisé combiné comprend une troisième bascule de type D (116) ayant son entrée D couplée de manière à recevoir les signaux synchronisés combinés des données d'échantillonnage et de la sélection de circuit, son entrée d'horloge couplée de manière à recevoir le signal d'horloge du dispositif périphérique et son entrée libre couplée à la sortie de la porte ET (108) et une seconde porte ET (120) ayant une première entrée couplée à la sortie de type Q de la bascule (116) et sa seconde entrée couplée à l'entrée de type D de la bascule (116).

6. Appareil selon la revendication 5, dans lequel ledit moyen (126) de sélection par une porte est un multiplexeur (126) ayant sa première entrée couplée à la sortie de la troisième bascule de type D (116) et une seconde entrée couplée à la sortie de la première porte ET mentionnée, ses entrées couplées de manière à recevoir les signaux des données d'échantillonnage (DS) et de sélection de circuit (CS) et sa sortie de sélection couplée pour recevoir le signal de mode (BMODE).

7. Appareil selon la revendication 5, dans lequel est placé un moyen pour utiliser le signal sélectionné par une porte comme signal de commande des dispositifs périphériques lire/écrire pour commander l'accès aux périphériques aux bus de données utilisés par le dispositif périphérique pour communiquer avec le système.

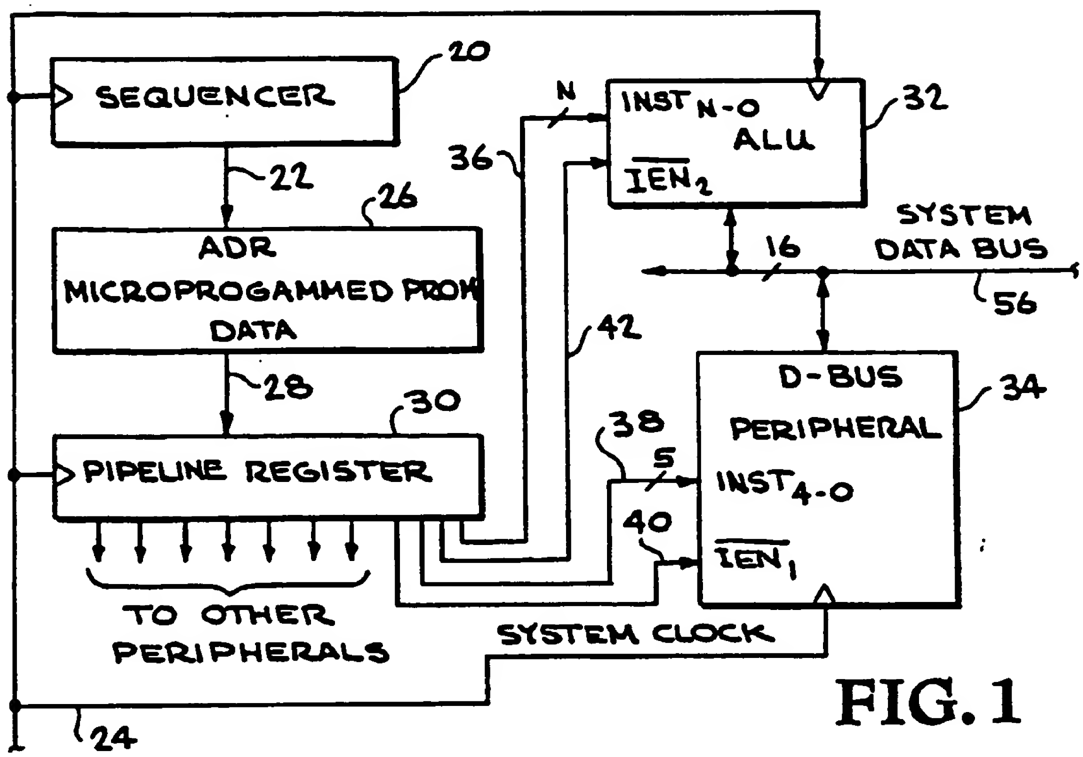

SEQUENCER
20
22
26
ADR
MICROPROGAMMED PROM
DATA
28
30
PIPELINE REGISTER
TO OTHER PERIPHERALS
24
N
INST N-O
ALU
IEN2
32
36
SYSTEM DATA BUS
16
56
42
D-BUS PERIPHERAL
34
38
5
INST4-O
40
IEN1
SYSTEM CLOCK

FIG. 1

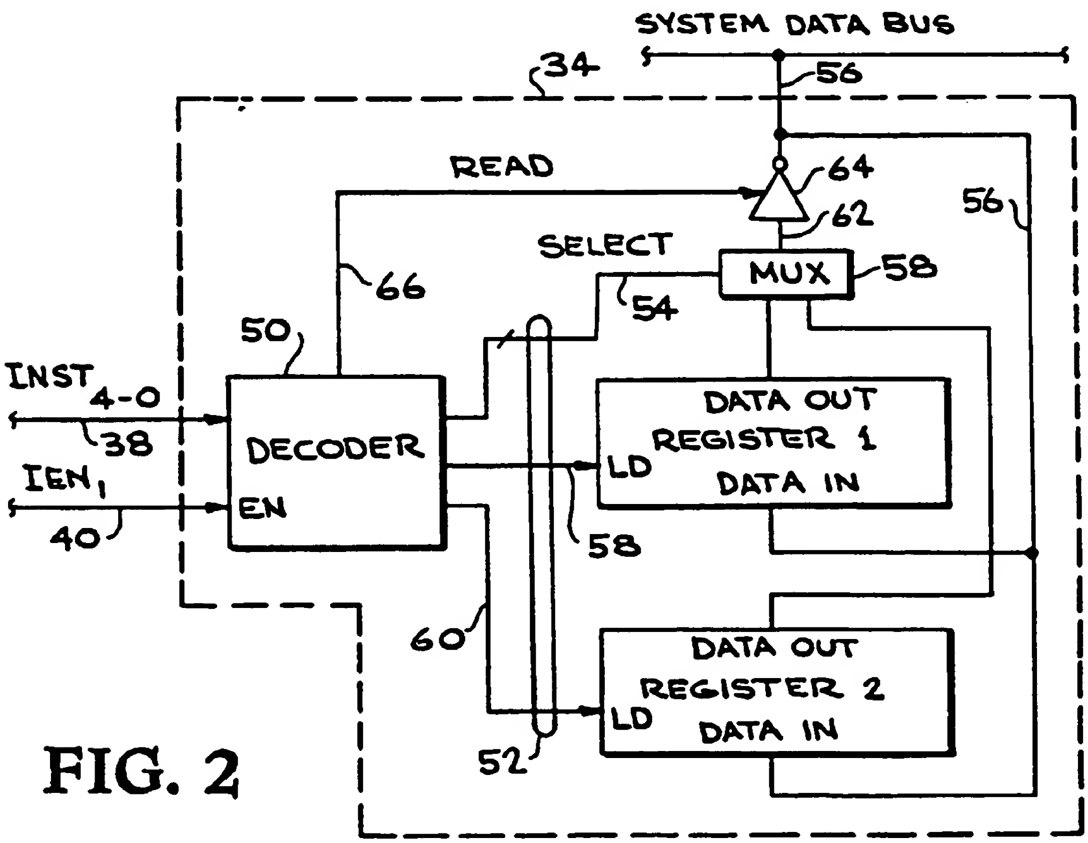

SYSTEM DATA BUS
34
56
READ
64
62
56
SELECT
MUX
58
66
54
50
INST 4-O
38
DECODER
IEN1
EN
40
DATA OUT
REGISTER 1
LD
DATA IN
58
60
DATA OUT
REGISTER 2
LD
DATA IN
52

FIG. 2

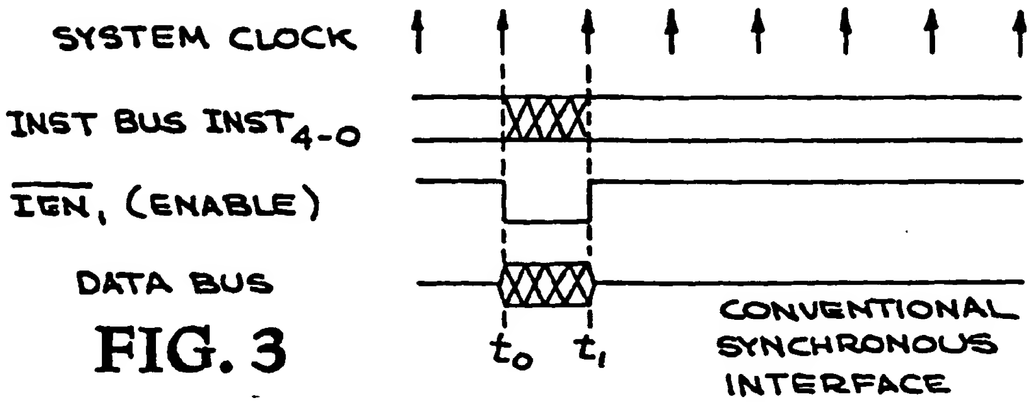


FIG. 3

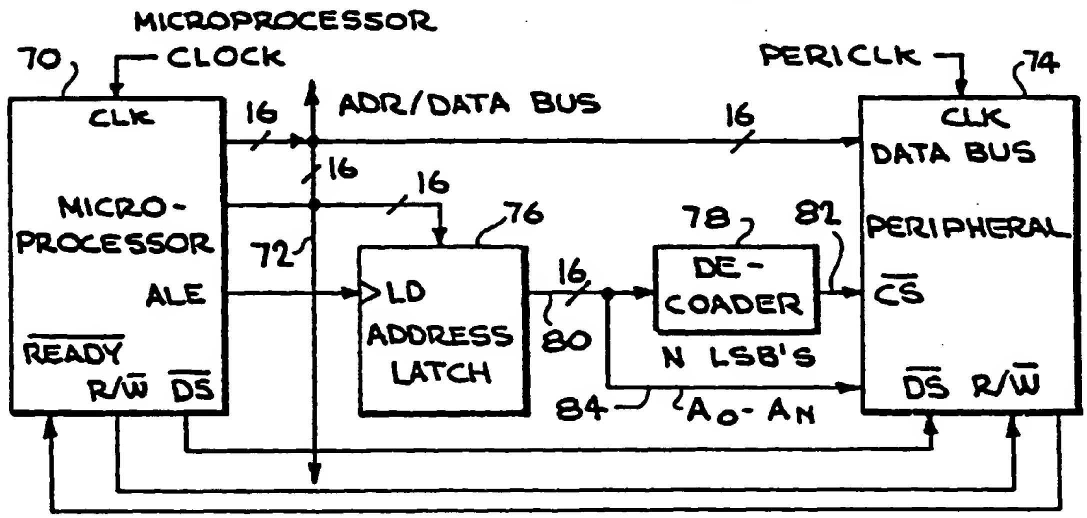


FIG. 4

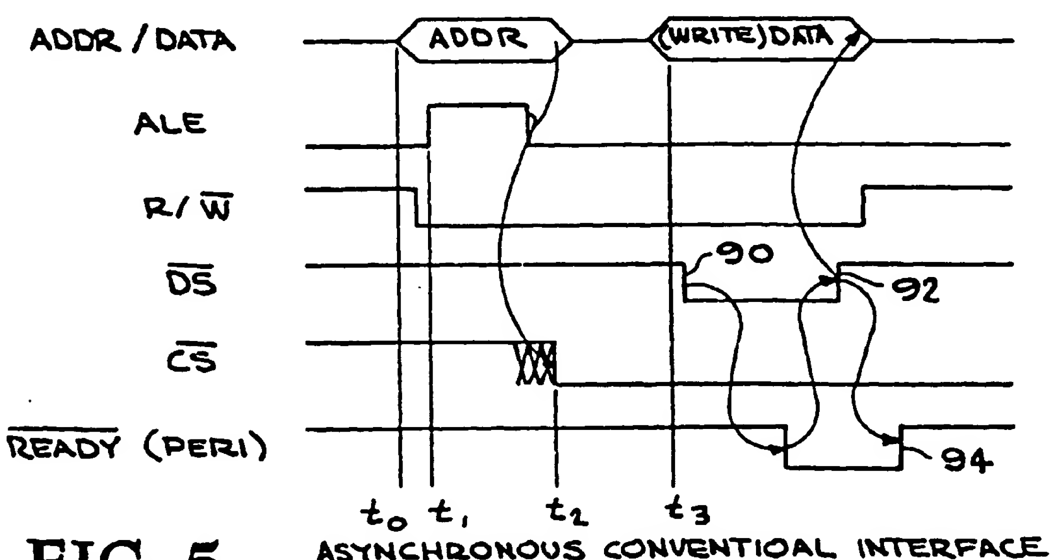


FIG. 5

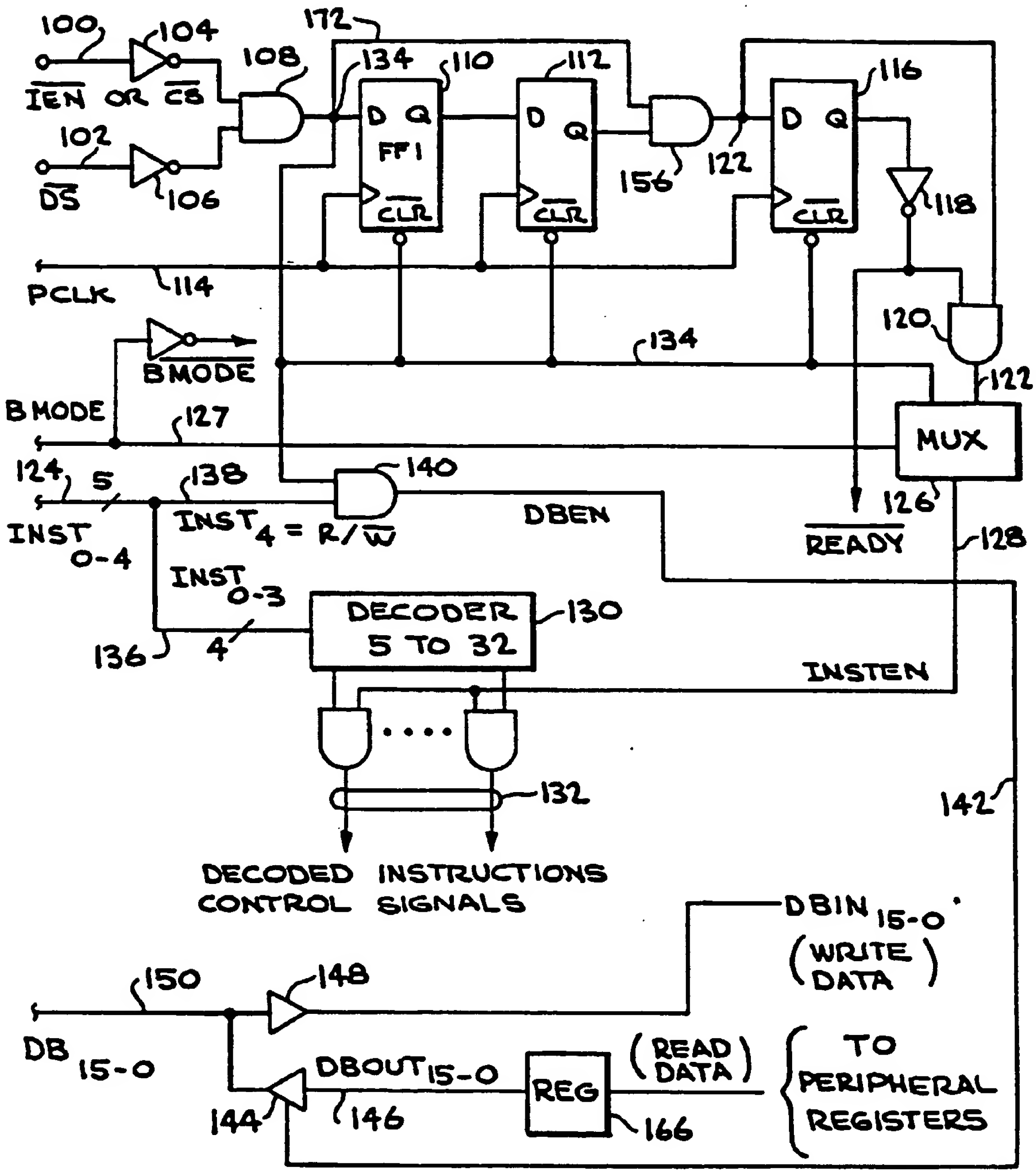

100
104
IEN OR CS
102
106
DS
108
172
134
110
112
D Q
FF1
CLR
D Q
CLR
156
122
116
D Q
CLR
118
PCLK
114
120
134
BMODE
BMODE
127
122
MUX
124
5
138
INST
0-4
140
$INST_4 = R/\overline{W}$
DBEN
126
READY
128
INST
0-3
DECODER
5 TO 32
130
136
4
INSTEN
132
142
DECODED INSTRUCTIONS
CONTROL SIGNALS
DBIN 15-0
(WRITE DATA)
150
148
DB 15-0
DBOUT 15-0
REG
(READ DATA)
TO PERIPHERAL REGISTERS
144
146
166

FIG. 6

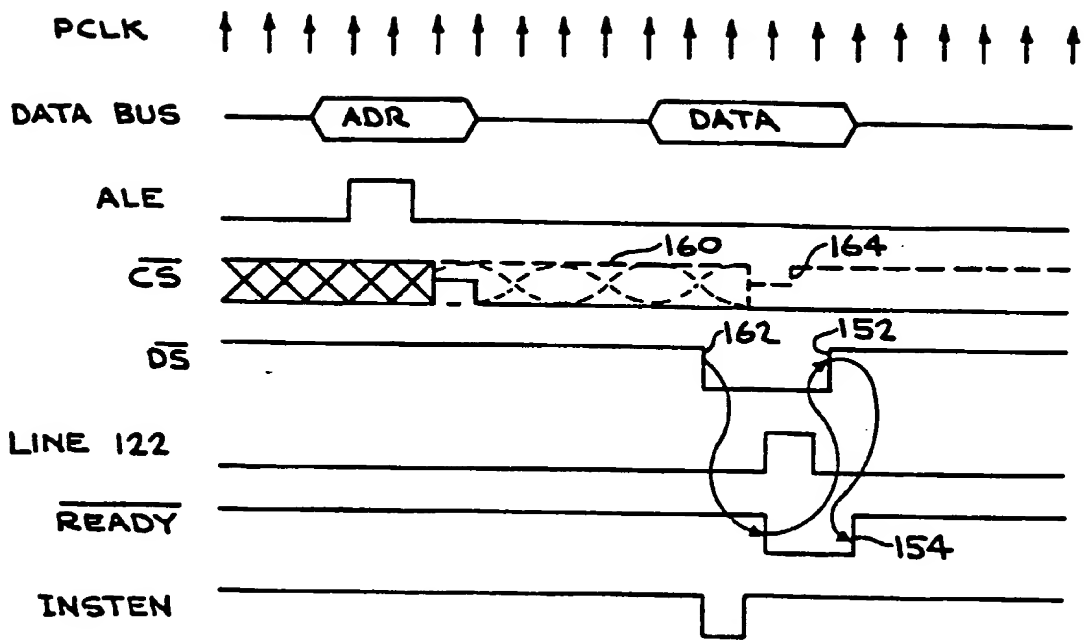
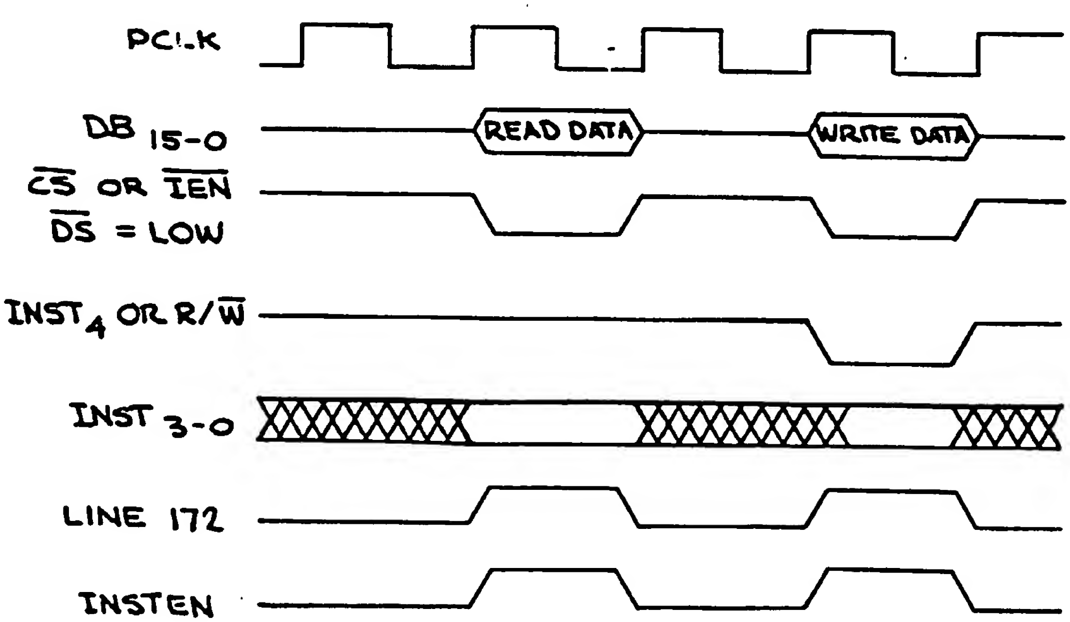


FIG. 7



READ AND WRITE CYCLE TIMING WITH A MICROPROGRAMMED PROCESSOR B MODE = 1

FIG. 9

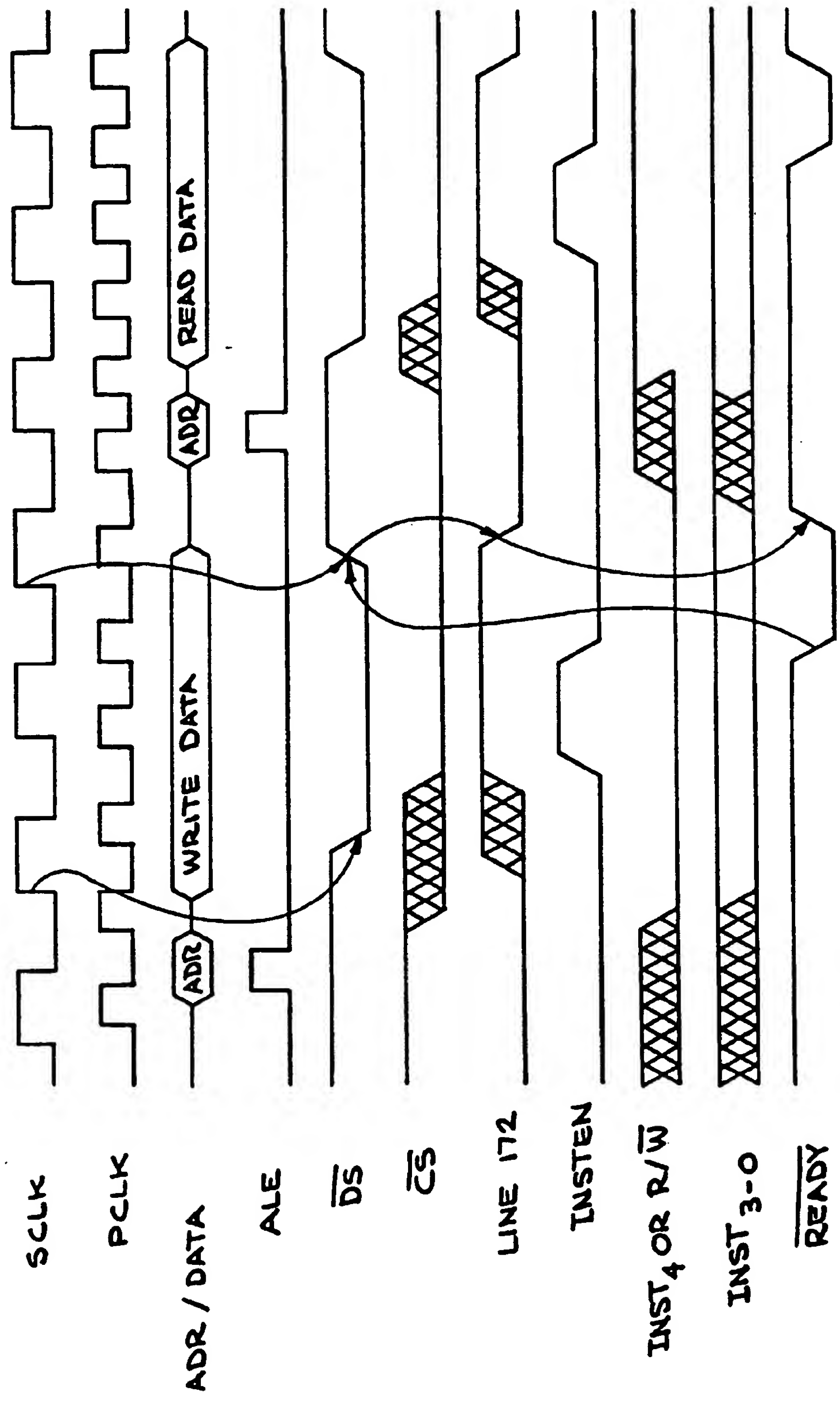


FIG. 8

WRITE AND READ CYCLE TIMING WHEN BMODE = 0 FOR AN ASYNCHRONOUS PROCESSOR

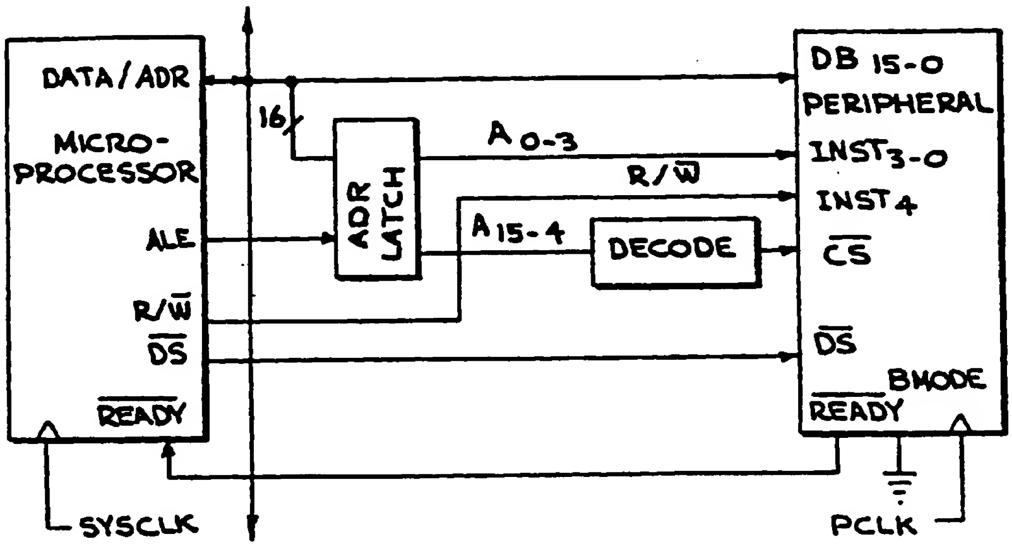

DATA / ADR
MICRO-PROCESSOR
ALE
R/W̄
D̄S̄
R̄ĒĀD̄Ȳ
16
ADR LATCH
A 0-3
R/W̄
A 15-4
DECODE
DB 15-0
PERIPHERAL
INST 3-0
INST 4
C̄S̄
D̄S̄
BMODE
R̄ĒĀD̄Ȳ
SYSCLK
PCLK

FIG. 10

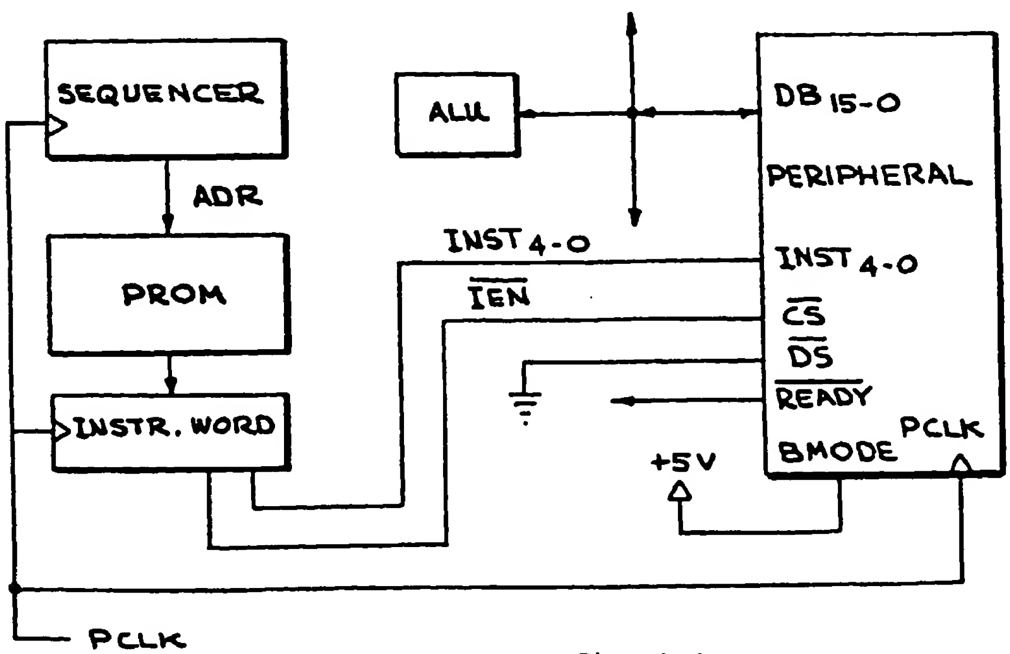

SEQUENCER
ADR
PROM
INSTR. WORD
ALU
INST 4-0
ĪĒN̄
DB 15-0
PERIPHERAL
INST 4-0
C̄S̄
D̄S̄
R̄ĒĀD̄Ȳ
BMODE
PCLK
+5V
PCLK

FIG. 11